

**15-121 Sample Assessment 3**

Up to 50 minutes. No calculators, no notes, no books, no computers. Show your work!

1. Consider the following program that creates a linked list. You may assume that the `ListNode` class exists and was defined as in class.

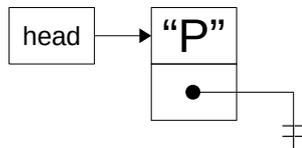
```

1 public class Quiz6CT1 {
2     public static void main(String[] args) {
3         ListNode<String> head = null;
4         ListNode<String> a = new ListNode<String>("Z");
5         ListNode<String> tmp = null;
6
7         head = new ListNode<String>("P");
8         tmp = new ListNode<String>("W");
9         tmp.next = a;
10        head.next = tmp;
11        tmp = new ListNode<String>("T");
12        tmp.next = head.next;
13        head.next = tmp;
14        tmp = tmp.next;
15        tmp.next = new ListNode<String>("S");
16        tmp.next.next = a;
17    }
18 }

```

Draw the state of the linked list after the execution of each specified line of code. (The linked list is defined as starting at `head`.) The first one is drawn for you.

- (a) After Line 7



- (b) (3 points) After Line 9

- (c) (3 points) After Line 10

- (d) (4 points) After Line 13

- (e) (4 points) After Line 16

2. Consider the following program that creates a linked list. You may assume that the `ListNode` class exists and was defined as in class.

```
1 public class Quiz6CT2 {
2     public static void main(String[] args) {
3         int[] nums = {44, 93, 87, 45, 57};
4         ListNode<Integer> head = new ListNode<Integer>(23);
5         ListNode<Integer> tmp = null;
6         ListNode<Integer> a = null;
7
8         for(int i = 0; i < nums.length; i++) {
9             tmp = new ListNode<Integer>(nums[i]);
10            tmp.next = head;
11            head = tmp;
12        }
13        a = tmp.next.next.next.next;
14        a.next.next = head;
15        a.next.next.next = null;
16        head = a;
17    }
18 }
```

Draw the state of the linked list after the execution of each specified line of code. (The linked list is defined as starting at `head`.)

(a) (2 points) After Line 12

---

(b) (2 points) After Line 15

---

(c) (2 points) After Line 16

### 3. Tickets, Get Your Tickets!

Imagine that you are working for FIFA on their ticket purchase website for the World Cup. Due to the extremely high demand for tickets, they have instituted a queuing system whereby customers need to electronically “wait in line” in order to login and buy tickets. You have been tasked with writing the data structure that will maintain this queue of customers.

You have been given the following specifications for the `TicketQueue` data type:

1. Customers should be served in the order that they enter the queue. So, if Joe enters the queue, and then later Fred enters the queue, Joe will be served before Fred.
2. It should have a method that adds a customer to the end of the queue.
3. It should have a method that determines which customer should be serviced next, removes that customer from the queue, and returns them. (This method needs to consider rule 1 above.)
4. It should have a method that returns how many customers are currently waiting.
5. It should have a method that returns the current place in line of a given customer. (The first customer in line is in place 1, the customer that will be served after them is in place 2, etc.)

- (a) (6 points) Write an interface, `TicketQueue`, based on the description above. (It will have four methods.) You can choose the method names yourself, but you should make sensible choices. In this part you are only writing an interface. You can assume that the class `Customer` already exists and stores information about a customer.

- (b) (14 points) Write a class (name it whatever you want) that implements the `TicketQueue` interface and functions as described above. Your implementation must use a singly linked list as the way it stores the data. For full credit, the methods should be all be  $O(1)$ , except the method that tells you a Customer's place in line: That method may be  $O(N)$ .

To save you time, you may assume that a `ListNode` class, similar to the homework, exists. You do not need to write it again in your answer. It is reprinted here in case you don't remember it:

```
public class ListNode<DataType> {
    public DataType data;
    public ListNode<DataType> next;

    public ListNode(DataType data) {
        this.data = data;
        this.next = null;
    }
}
```

Hints:

- We are not specifying how you should handle error cases. Handle them in any suitable way.
- Make a plan for how you will store your data before you start writing code. How you will meet the efficiency requirement should be a part of this plan.
- You are writing an entire class, so make sure to include a proper prototype.

Additional space.